

MSISpIC: A Probabilistic Scan Matching Algorithm Using a Mechanical Scanned Imaging Sonar

Emili Hernández, Pere Ridao, David Ribas and Joan Batlle

Abstract—This paper compares two well known scan matching algorithms: the MbICP and the pIC. As a result of the study, it is proposed the MSISpIC, a probabilistic scan matching algorithm for the localization of an Autonomous Underwater Vehicle (AUV). The technique uses range scans gathered with a Mechanical Scanning Imaging Sonar (MSIS), and the robot displacement estimated through dead-reckoning with the help of a Doppler Velocity Log (DVL) and a Motion Reference Unit (MRU). The proposed method is an extension of the pIC algorithm. Its major contribution consists in: 1) using an EKF to estimate the local path traveled by the robot while grabbing the scan as well as its uncertainty and 2) proposing a method to group into a unique scan, with a convenient uncertainty model, all the data grabbed along the path described by the robot. The algorithm has been tested on an AUV guided along a 600m path within a marina environment with satisfactory results.

Index Terms—Autonomous robots.

I. INTRODUCTION

DURING a long term mission with an autonomous robot it is necessary to keep the track of the vehicle's position. Scan matching is a technique that can be used to estimate the vehicle displacement using successive range scans. Many applications in robotics like mapping, localization or tracking use this technique to estimate the robot's relative displacement [1], [2], [3] (to mention some but a few). Scan Matching estimates the robot relative displacement between two configurations, by maximizing the overlap between the range scans normally gathered with a laser or a sonar sensor. Moreover, the scans can then be mapped to set up a local map of the surrounding environment of the robot to be used for reactive/deliberative obstacle avoidance like in [4].

The existing scan matching techniques can be divided in two groups depending on if they use high-level entities like lines or planes or otherwise they rely on the raw scan. On one hand, it is possible to assume the existence of polygonal structures in structured environments as is supposed in [5] [6] [7], or even in some underwater applications [8]. However, extracting simple polygons for representing the environment is not always possible, particularly in unstructured scenarios which are the most common in underwater robotics. On the other hand, there is a second type of algorithms that work with raw data from the scanner to solve the matching. Usually, these techniques are based on a two step iterative process which is repeated till convergence. The sensor displacements are computed by approximating the solution to the best overlap between two scans by looking for the closest point

for each single data of the scan. After that, a minimization process to estimate the solution is done. The process is repeated until convergence. The most popular technique is the ICP (Iterative Closest Point) algorithm [9] which has been modified in different ways [10]. Most of the versions of the ICP algorithm use the Euclidian distance to estimate the correspondences between scans. However, this distance does not take into account that small rotations of the sensor mean large displacements as the distance is increased. To overcome this limitation several approaches have been done. The IDC (Iterative Dual Correspondence) algorithm [11] computes two different minimizations, one for the translation and another for the rotation. The MbICP (Metric-based Iterative Closest Point) algorithm [12] establishes a new distance concept which captures the sensor displacement and rotation at the same time. However, none of these algorithms take into account the sensor nor the displacement uncertainties which are very important, specially when sonar sensors are used. The probabilistic Iterative Correspondence method (pIC), proposed in [13], explicitly deals with those uncertainties to decide which points in the reference scan are statistically compatible with a certain point of the new scan. Moreover, the minimization process is carried out over the Mahalanobis distance. Although in this case the algorithm was applied to laser scans, its probabilistic formulation is able to improve the robustness and accuracy. On the other hand, in [14] another probabilistic approach is proposed in order to deal with the noisy and sparse measurements obtained from conventional time-of-fly sonar sensors of a mobile robot. This method deals with the sparsity of readings by grouping sonar measurements along short trajectories. It uses probabilistic models of ultrasonic and odometry sensors as well as a method to propagate the error through them in order to estimate a group of scan positions together with their uncertainty.

This work presents a study where the MbICP and pIC algorithms are compared to deal with acoustic data. Then we propose the MSISpIC as an extension of the basic pIC algorithm to deal with data gathered by an AUV using a MSIS. In our approach, the robot moves while scanning the environment. Hence, an EKF using a constant velocity model with acceleration noise, updated with velocity and attitude measurements obtained from a DVL and a MRU respectively, is used to estimate the trajectory followed by the robot along the scan. This trajectory is used to remove the motion induced distortion of the acoustic image as well as to predict the uncertainty of the range scans prior to register them through the standard pIC algorithm.

The paper is structured as follows. In section II it is described the metric scan matching using the MbICP algorithm.

The authors are with the Dept. Computer Engineering, University of Girona, 17071 Girona, Spain.
E-mail: {emilihb,pere,drivas,jbatlle}@eia.udg.edu

Section III shows the probabilistic scan matching algorithm. Next, in section IV the two algorithms are compared using synthetic noisy data. Section V details the process to grab complete scans from the MSIS to be used in our scan matching algorithm which is described in section VI. Section VII reports the experimental results using the MSISpIC before conclusions.

II. METRIC SCAN MATCHING

In the conventional ICP algorithm correspondences between two scans are chosen based on the closest-point rule, normally using the Euclidean distance. As pointed out in [13], this distance does not take into account that the points in the new scan, which are far from the sensor, could be far from their correspondences in the previous scan due to a rotation. To overcome this limitation, the MbICP algorithm [12] proposes a new concept of distance using the translation and rotation simultaneously. Introducing this new distance into the ICP framework, translational and rotational movements are taken into account at once while looking for the correspondences as well as during the minimization process.

Given a certain displacement vector $\hat{\mathbf{q}} = (\hat{x}, \hat{y}, \hat{\theta})$ where $\hat{\theta} \in [-\pi, \pi]$, a norm function can be defined as:

$$\|\hat{\mathbf{q}}\| = \sqrt{\hat{x}^2 + \hat{y}^2 + L^2 \hat{\theta}^2}$$

where L is a positive real number.

Then, given two points, $\mathbf{p}_1 = (p_{1x}, p_{1y})$ and $\mathbf{p}_2 = (p_{2x}, p_{2y})$ in \mathbb{R}^2 , a distance function can be defined as follows:

$$d(\mathbf{p}_1, \mathbf{p}_2) = \|\arg \min_{\hat{\mathbf{q}}} \|\mathbf{p}_2 - \hat{\mathbf{q}} \oplus \mathbf{p}_1\|\|$$

As a distance it accomplishes the following properties:

- 1) Positiveness: $d(\mathbf{p}_1, \mathbf{p}_2) \geq 0$
- 2) Symmetry: $d(\mathbf{p}_1, \mathbf{p}_2) = d(\mathbf{p}_2, \mathbf{p}_1)$
- 3) Triangular inequality: $d(\mathbf{p}_1, \mathbf{p}_3) \leq d(\mathbf{p}_1, \mathbf{p}_2) + d(\mathbf{p}_2, \mathbf{p}_3)$
- 4) $d(\mathbf{p}_1, \mathbf{p}_2) = 0 \Rightarrow \mathbf{p}_1 = \mathbf{p}_2$

This distance is used in the ICP framework becoming the basis of the MbICP algorithm (see Algorithm 1). The inputs are the reference scan S_{ref} with points \mathbf{r}_i ($i = 1..n$), the new scan S_{new} with points \mathbf{n}_j ($j = 1..m$) and the initial relative displacement estimation $\hat{\mathbf{q}}$. The following procedure is iteratively executed until convergence. First, the points of the new scan (\mathbf{n}_j) are compounded with the robot displacement ($\hat{\mathbf{q}}$) as follows:

$$\mathbf{c}_j = \hat{\mathbf{q}} \oplus \mathbf{n}_j = \begin{pmatrix} \hat{x} + n_{jx} \cos \hat{\theta} - n_{jy} \sin \hat{\theta} \\ \hat{y} + n_{jx} \sin \hat{\theta} + n_{jy} \cos \hat{\theta} \end{pmatrix} \quad (1)$$

Assuming that the rotation between both scans is small enough, the following simplification is applied:

$$\sin \hat{\theta} \approx \hat{\theta}; \cos \hat{\theta} \approx 1$$

obtaining a linear approximation of the eq. 1:

$$\mathbf{c}_j = \hat{\mathbf{q}} \oplus' \mathbf{n}_j = \begin{pmatrix} \hat{x} + n_{jx} - \hat{\theta} n_{jy} \\ \hat{y} + \hat{\theta} n_{jx} + n_{jy} \end{pmatrix} \quad (2)$$

The result (\mathbf{c}_j), are the points of the new scan referenced to the reference frame. Then, for each point (\mathbf{c}_j) it is possible to compute its association point (\mathbf{a}_j) using the Closest-Point Rule because as stated in eq. 2, the relation between \mathbf{n}_j and \mathbf{c}_j is a linear function of θ and it can be proved (see [12] for details) that the square of their norm is a quadratic equation in θ . Solving the equation to find the θ that minimizes the norm allows to find the distance function used in the MbICP algorithm:

$$d(\mathbf{r}_i, \mathbf{c}_j) = \sqrt{\delta_x^2 + \delta_y^2 - \frac{(\delta_x r_{iy} - \delta_y r_{ix})^2}{r_{iy}^2 + r_{ix}^2 + L^2}} \quad (3)$$

where

$$\begin{aligned} \delta_x &= c_{jx} - r_{ix} \\ \delta_y &= c_{jy} - r_{iy} \end{aligned}$$

Once the correspondences \mathbf{a}_j have been found, it is possible to estimate the displacement $\hat{\mathbf{q}}_{min}$ which minimizes the mean square error between \mathbf{a}_j and \mathbf{c}_j using the eq. 3 distance function. This is done using Least Squares minimization (eq. 4). If there is convergence, the function returns, otherwise another iteration is required.

$$q_{min} = \arg \min_{\mathbf{q}} \left\{ \sum_{j=1}^n \left(\delta_{jx}^2 + \delta_{jy}^2 - \frac{(\delta_{jx} a_{jy} - \delta_{jy} a_{jx})^2}{a_{jy}^2 + a_{jx}^2 + L^2} \right) \right\} \quad (4)$$

where

$$\begin{aligned} \delta_{jx} &= c_{jx} - c_{jy} \theta + x - a_{jx} \\ \delta_{jy} &= c_{jx} \theta + c_{jy} + y - a_{jy} \end{aligned}$$

Algorithm 1 The MbICP algorithm

```

 $\hat{\mathbf{q}}_{MbICP} = MbICP(S_{ref}, S_{new}, \hat{\mathbf{q}})$  {
   $k = 0$ 
   $\hat{\mathbf{q}}_k = \hat{\mathbf{q}}$ 
  do {
    for( $j = 0; size(S_{new}); j++$ )
       $\mathbf{a}_j = \arg \min_{\mathbf{r}_i} \{d(\mathbf{r}_i, \hat{\mathbf{q}}_k \oplus \mathbf{n}_j)\}, \forall \mathbf{r}_i \in S_{ref} (eq. 3)$ 
     $\hat{\mathbf{q}}_{min} = \arg \min_{\mathbf{q}} \left\{ \sum_j ((\mathbf{a}_j - \mathbf{q} \oplus \mathbf{n}_j)^2) \right\}$ 
    if(Convergence())
       $\hat{\mathbf{q}}_{MbICP} = \hat{\mathbf{q}}_{min}$ 
    else {
       $\hat{\mathbf{q}}_{k+1} = \hat{\mathbf{q}}_{min}$ 
       $k++$ 
    }
  }
  while(!Convergence() and  $k < maxIterations$ )
}
```

III. PROBABILISTIC SCAN MATCHING

Despite the increase of accuracy and robustness of the MbICP algorithm, it does not model the uncertainty of the sensor measurements. Because of that, if the scan data is very noisy, two statistically compatible points could appear far enough, in terms of the Euclidean distance. This situation

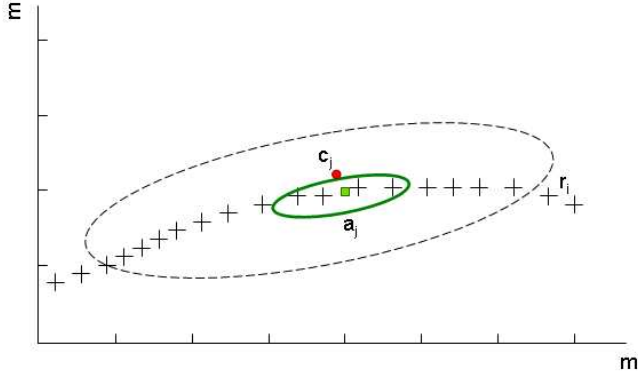


Fig. 1. pIC correspondence computation. The large ellipse contains all the statistically compatible points and the squared point represents the correspondence with its uncertainty (small ellipse).

might prevent a possible association or even generate a wrong one. The pIC algorithm proposed in [13] is a statistical extension of the ICP algorithm where the relative displacement \mathbf{q} as well as the observed points in both scans \mathbf{r}_i and \mathbf{n}_j , are modeled as random Gaussian variables (r.g.v.). Whereas geometric ICP algorithm uses the closest-point rule to find the correspondence for a point in the new scan, the pIC algorithm first computes the set of compatible points (in terms of the Mahalanobis distance) and then computes the virtual 'expected' compatible point to be used as the correspondence (Fig. 1).

For a better understanding the algorithm is reproduced here (Algorithm 2). The inputs are the reference scan S_{ref} with points \mathbf{r}_i ($i = 1..n$), the new scan S_{new} with points \mathbf{n}_j ($j = 1..m$) and the initial relative displacement estimation $\hat{\mathbf{q}}$ with its covariance \mathbf{P}_q . The following procedure is iteratively executed until convergence. First, the points of the new scan (\mathbf{n}_j) are compounded with the robot displacement (\mathbf{q}_k). The result (\mathbf{c}_j), are the points of the new scan referenced to the reference frame. Then, for each point (\mathbf{c}_j), a set (A_j) of all the compatible points in the reference scan (S_{ref}) is established using a compatibility test over the Mahalanobis distance. Next step consists in computing the virtual association point (\mathbf{a}_j) as the expectancy over the random variable defined by the set (A_j). It is worth noting that if $\mathbf{q} \equiv N(\hat{\mathbf{q}}, \mathbf{P}_q)$, $\mathbf{n}_j \equiv N(\hat{\mathbf{n}}_j, \mathbf{P}_{n_j})$ and $\mathbf{r}_i \equiv N(\hat{\mathbf{r}}_i, \mathbf{P}_{r_i})$, it is possible to compute the probability $p(\mathbf{r}_i = \mathbf{c}_j)$ for every element (\mathbf{r}_i) of A_j of being a correct matching for \mathbf{c}_j . In order to do so, let us define a r.g.v. which describes the error of the $\{\mathbf{r}_i, \mathbf{c}_j\}$ pairing:

$$\mathbf{e}_{ij} = \mathbf{r}_i - \mathbf{q}_k \oplus \mathbf{n}_j ; \mathbf{e}_{ij} \approx N(\hat{\mathbf{r}}_i - \hat{\mathbf{q}}_k \oplus \hat{\mathbf{n}}_j, \mathbf{P}_{e_{ij}})$$

$$\mathbf{P}_{e_{ij}} = \mathbf{P}_{r_i} + \mathbf{J}_q \mathbf{P}_q \mathbf{J}_q^T + \mathbf{J}_n \mathbf{P}_{n_j} \mathbf{J}_n^T$$

with

$$\mathbf{J}_q = \left. \frac{\partial \mathbf{r}_i - \mathbf{q} \oplus \mathbf{n}_j}{\partial \mathbf{q}} \right|_{\hat{\mathbf{q}}}, \mathbf{J}_n = \left. \frac{\partial \mathbf{r}_i - \mathbf{q} \oplus \mathbf{n}_j}{\partial \mathbf{n}_j} \right|_{\hat{\mathbf{n}}_j}$$

then $p(\mathbf{r}_i = \mathbf{c}_j) = p(\mathbf{e}_{ij} = 0)$ where $p(\mathbf{e}_{ij} = 0)$ can be computed as follows:

$$p(\mathbf{e}_{ij} = 0) = \frac{f_{e_{ij}}(\mathbf{r}_i - \mathbf{c}_j)}{\sum_{\mathbf{r}_i \in A_j} f_{e_{ij}}(\mathbf{r}_i - \mathbf{c}_j)} \quad (5)$$

where $f_{e_{ij}}$ is the probability density function of \mathbf{e}_{ij} r.g.v. Once $\hat{\mathbf{a}}_j$ has been computed, a similar procedure can be used to estimate its uncertainty \mathbf{P}_{a_j} , before computing the uncertainty \mathbf{P}_{e_j} of the matching error ($\hat{\mathbf{a}}_j - \hat{\mathbf{c}}_j$). Now, it is possible to estimate the displacement $\hat{\mathbf{q}}_{min}$ which minimizes the mean square error of the *Mahalanobis Distance* [15] between $\hat{\mathbf{a}}_j$ and $\hat{\mathbf{c}}_j$. This is done using Least Squares minimization method. If there is convergence, the function returns, otherwise another iteration is required.

Algorithm 2 The pIC algorithm

```

 $\hat{\mathbf{q}}_{pIC} = pIC(S_{ref}, S_{new}, \hat{\mathbf{q}}, \mathbf{P}_q) \{$ 
   $k = 0$ 
   $\hat{\mathbf{q}}_k = \hat{\mathbf{q}}$ 
  do {
    for( $j = 0; size(S_{new}); j++$ ) {
       $\hat{\mathbf{c}}_j = \hat{\mathbf{q}}_k \oplus \hat{\mathbf{n}}_j$ 
       $A_j = \{\mathbf{r}_i \in S_{ref} / D_M^2(\mathbf{r}_i, \mathbf{c}_j) \leq \chi_{2,\alpha}^2\}$ 
       $\hat{\mathbf{a}}_j = \sum_j \hat{\mathbf{r}}_i p(\mathbf{r}_i = \mathbf{c}_j), \forall \mathbf{r}_i \in A_j$ 
       $\mathbf{P}_{a_j} = \sum_{\mathbf{r}_i \in A_j} [(\hat{\mathbf{r}}_i - \hat{\mathbf{a}}_j)(\hat{\mathbf{r}}_i - \hat{\mathbf{a}}_j)^T + \mathbf{P}_{r_i}] p(\mathbf{r}_i = \mathbf{c}_j)$ 
       $\mathbf{P}_{e_j} = \mathbf{P}_{a_j} + \mathbf{J}_q \mathbf{P}_q \mathbf{J}_q^T + \mathbf{J}_n \mathbf{P}_{n_j} \mathbf{J}_n^T$ 
    }
     $\hat{\mathbf{q}}_{min} = \arg \min_{\mathbf{q}} \left\{ \sum_j ((\hat{\mathbf{a}}_j - \hat{\mathbf{c}}_j)^T \mathbf{P}_{e_j}^{-1} (\hat{\mathbf{a}}_j - \hat{\mathbf{c}}_j)) \right\}$ 
    if(Convergence())
       $\hat{\mathbf{q}}_{pIC} = \hat{\mathbf{q}}_{min}$ 
    else {
       $\hat{\mathbf{q}}_{k+1} = \hat{\mathbf{q}}_{min}$ 
       $k++$ 
    }
  }
  while(!Convergence() and  $k < maxIterations$ )
}
```

IV. COMPARISON BETWEEN MBICP AND PIC ALGORITHMS

In order to examine in detail the effectiveness of the MbICP and the pIC algorithms, several tests with synthetic data were carried out. Simulated data was generated with Matlab using two different state vectors \mathbf{q}_1 and \mathbf{q}_2 as an input for a script to generate a static scan of a rectangular scenario. The initial guess for the displacement \mathbf{q}_0 was computed using the real displacement of the robot with Gaussian additive noise.

Two virtual scans, depicted in Fig. 2 without noise, have been used as a reference dataset to test the two techniques compared in this paper. In both cases, it is assumed that the robot was stopped when the scan was gathered. Hence, all the noise comes from the scanner sensor, so the scans are not distorted except for the used noise level, which is more or less the same as found in commercial MSIS sensors. This case is quite similar to the conventional case in which laser scan

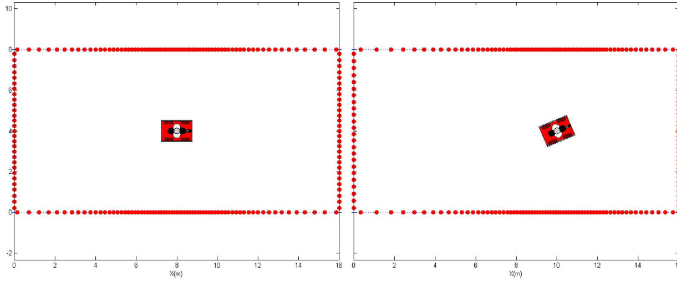


Fig. 2. Matlab virtual scans without noise: a) Scan k; b) Scan k+1.

Alg. \ Noise	$\sigma_r = 0.05$ $\sigma_\psi = 1.5$	$\sigma_r = 0.1$ $\sigma_\psi = 3$	$\sigma_r = 0.2$ $\sigma_\psi = 8$
MbICP	(1.83, 0.68, 25.06)	(2.0, 25.78)	(1.94, -0.07, 15.77)
pIC	(1.97, 0.22.9)	(1.98, 0.02, 22.1)	(1.98, 0.08, 22.15)

TABLE I
DISPLACEMENT ESTIMATIONS. UNITS OF σ_r AND σ_ψ ARE METERS AND DEGREES RESPECTIVELY.

sensors are used. The displacement between the scans is $\mathbf{q}_0 = (2m, 0m, 22.5^\circ)$. At each execution, random noise is added to the range and bearing of each beam. In order to stabilize the results, each algorithm have been executed 50 times for each level of noise. Table I shows the means of the displacement estimations, Fig. 3 shows the mean of iterations needed to reach convergence for each noise step and Fig. 4 depicts the graphical solutions of a single representative execution.

According to Fig. 4, the MbICP algorithm is able to perform a good scan matching for scans with low noise. However, the results are not so good when the noise is increased. In Fig. 3, it can be appreciated that the number of iterations needed for the MbICP algorithm to reach convergence increases with respect to the applied uncertainty. This algorithm is computationally efficient because it only takes into account geometric information. However, it does not behave well in some situations where associations are not computed correctly, for example in

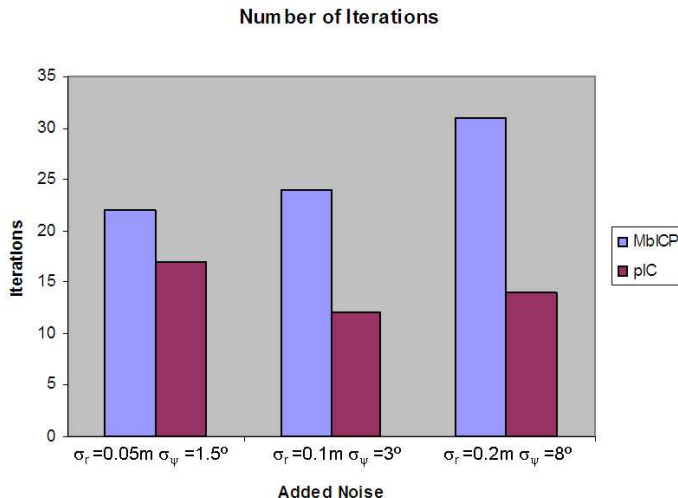


Fig. 3. Mean of iterations until convergence.

the corners (see Fig. 4 when $\sigma_\psi = 3^\circ$ and $\sigma_\psi = 8^\circ$). Hence, it is not a strong candidate to be used with uncertain sonar data.

In contrast, the pIC algorithm performs well in situations where the MbICP produces poor results. This technique does not present problems in associating data from corners because correspondences are virtual points created from statistically compatible points of the first scan. Thus, the algorithm is more robust and accurate (Table I). As shown in Fig. 3, the pIC algorithm requires fewer iterations to converge even when the noise is increased but the computational load of each iteration is higher. In addition, it also offers better results than the MbICP under the same conditions (Fig. 4). In conclusion, although the increased computational load due to the statistical nature of the algorithm, the pIC algorithm is a better candidate to be used with acoustic data. For this reason, the work that follows is based only on the pIC algorithm.

V. SCAN GRABBING USING A MSIS

Scan matching techniques are conceived to accept as input parameters two range scans with a rough displacement estimation between them. Most of the algorithms use laser range finders which gather scans almost instantaneously. However, for the underwater environment, commercially available scan sensors are based on acoustics. Most of these sensors have a mechanical head that rotates at fixed angular steps. At each step, a beam is emitted and received a posteriori, measuring ranges to the obstacles found across its trajectory. Hence, getting a complete scan lasts few seconds while the vehicle is moving, generating deformed scans. Therefore, a correction taking into account the robot pose when the beam was grabbed is necessary.

A. Beam segmentation and range detection

The MSIS returns a polar acoustic image composed of beams. Each beam has a particular bearing angle value and a set of intensity measurements. The angle corresponds to the orientation of the sensor head when the beam was emitted. The acoustic linear image corresponding to one beam is returned as an array of bins. A bin is an element of the array that represents an acoustic intensity detected at a certain distance with a value between 0 and 255 (Fig. 5). The beam is then segmented using a predefined threshold to compute the intensity peaks. Due to the noisy nature of the acoustic data, a minimum distance between peaks criteria is also applied. Hence, positions finally considered are those corresponding to high intensity values above the threshold with a minimum *distance* between each other. Fig. 6 illustrates this process.

B. Relative vehicle localization

The pIC algorithm needs a complete scan to be registered with the previous one in order to estimate the robot's displacement. Since MSIS needs a considerable period of time to obtain a complete scan, if the robot does not remain static, the robot's motion induces a distortion in the acoustic image (Fig. 7). To deal with this problem it is necessary to know the

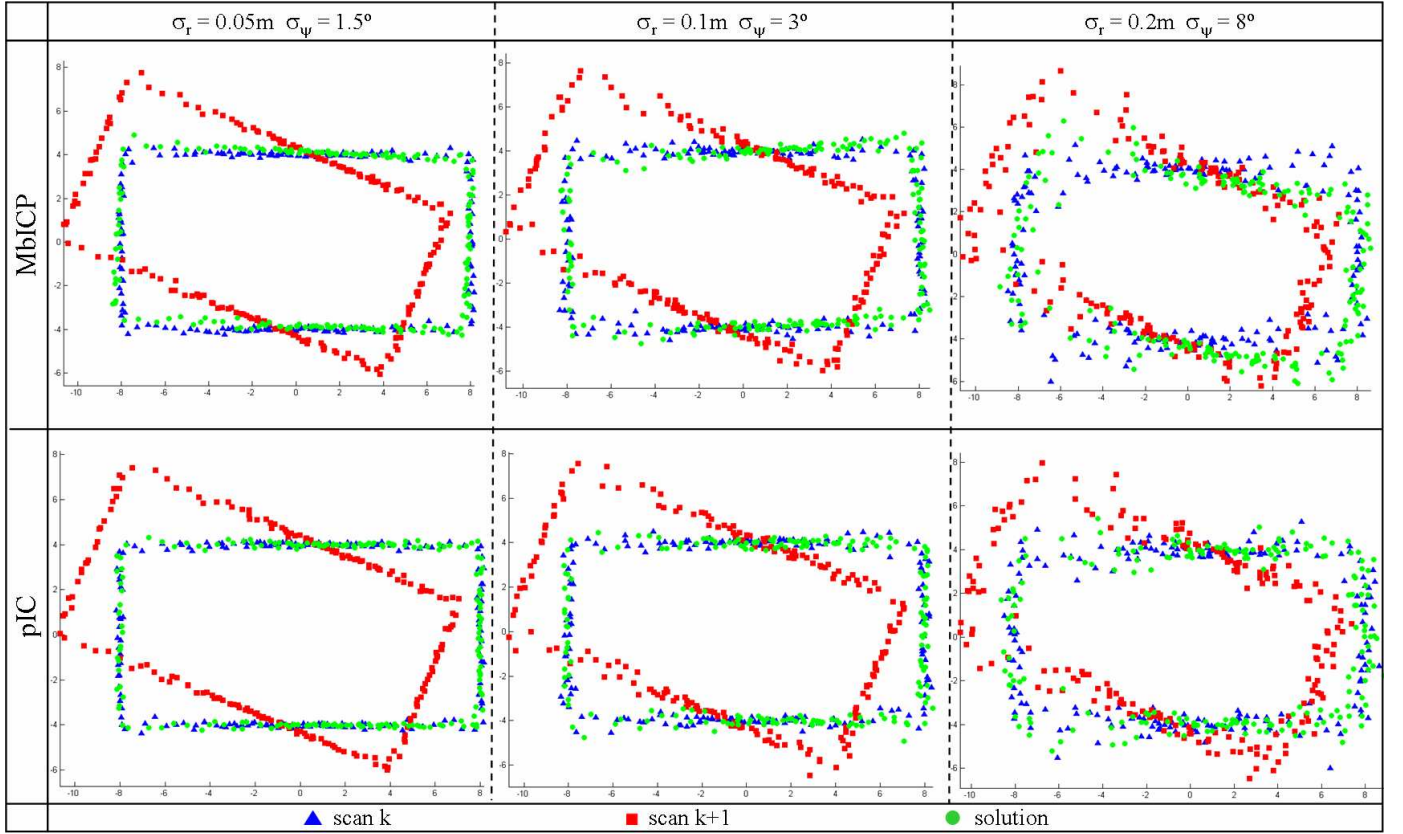


Fig. 4. MbICP and pIC results.

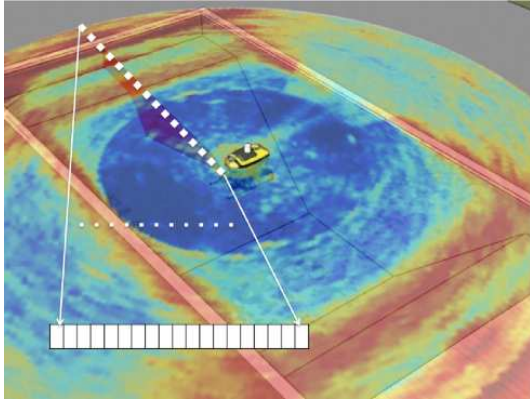


Fig. 5. Interpretation of a polar image gathered with an MSIS. The current beam is detailed.

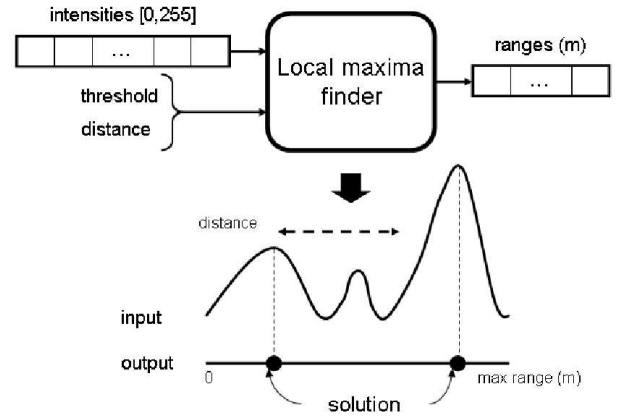


Fig. 6. Peaks detector for a MSIS beam.

robot's pose at the beam reception time. Hence, it is possible to define an initial coordinate system I to reference all the range measurements belonging to the same scan. In our case, this initial frame is fixed at the robot pose where the first beam of the current scan was read.

The localization system used in this work is a slight modification of the navigation system described in [16]. In this system, a Xsense MRU provides heading measurements and a SonTek Argonaut DVL unit which includes 2 inclinometers and a depth sensor is used to estimate the robot's pose during the scan (navigation problem). MSIS beams are read at 30

Hz while DVL and MTi readings arrive asynchronously at a frequency of 1.5 Hz and 10 Hz respectively. An EKF is used to estimate the robot's 6DOF pose whenever a sonar beam is read. DVL and MTi readings are used asynchronously to update the filter. To reduce the noise inherent to the DVL measurements, a simple 6DOF constant velocity kinematics model is used.

The information of the system at step k is stored in the state vector \mathbf{x}_k with estimated mean $\hat{\mathbf{x}}_k$ and covariance \mathbf{P}_k :

$$\hat{\mathbf{x}}_k = [\hat{\eta}^B, \hat{\nu}^R]^T \quad \mathbf{P}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \quad (6)$$

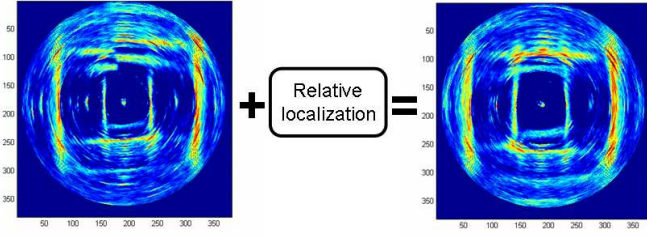


Fig. 7. Relative displacement correction.

with:

$$\eta^B = [x, y, z, \phi, \theta, \psi]^T; \quad \nu^R = [u, v, w, p, q, r]^T \quad (7)$$

where, as defined in [17], η^B is the position and attitude vector referenced to a base frame B , and ν^R is the linear and angular velocity vector referenced to the robot's coordinate frame R . The coordinate frame B is chosen coincident with I but oriented to the north, hence the compass measurements can be integrated straight forward.

The vehicle's movement prediction is performed using the 6DOF kinematic model:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) = \begin{bmatrix} \eta_k^B \\ \nu_k^R \end{bmatrix} = \begin{bmatrix} \eta_{k-1}^B + J(\eta_{k-1}^B) \nu_{k-1}^R T \\ \nu_{k-1}^R \end{bmatrix} \quad (8)$$

$$J(\eta) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi & 0 & 0 & 0 \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - s\psi s\phi & 0 & 0 & 0 \\ -s\theta & c\theta s\phi & c\theta c\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & s\phi t\theta & c\phi t\theta \\ 0 & 0 & 0 & 0 & c\phi & -s\phi \\ 0 & 0 & 0 & 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \quad (9)$$

Although in this model the velocity is considered to be constant, in order to allow for slight changes, a velocity perturbation modeled as the integral of a stationary white noise v_k is introduced. The covariance matrix \mathbf{Q}_k of this acceleration noise is diagonal and in the order of magnitude of the maximum acceleration increment that the robot may experience over a sample period.

$$\nu_k^R = \hat{\nu}_k^R + v_k T \quad (10)$$

$$E[v_k] = 0; \quad E[v_k v_j^T] = \delta_{kj} \mathbf{Q} \quad (11)$$

Hence, v_k is an acceleration noise which is integrated and is added in velocity (Equation 10), being nonlinearly propagated to the position. Finally, the model prediction and update is carried out as detailed below:

1) *Prediction*: The estimate of the state is obtained as:

$$\hat{\mathbf{x}}_k = f(\hat{\mathbf{x}}_{k-1}) \quad (12)$$

and its covariance matrix as:

$$\mathbf{P}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^T \quad (13)$$

where \mathbf{F}_k and \mathbf{G}_k are the Jacobian matrices of partial derivatives of the non-linear model function f with respect to the state $\mathbf{x}_{R,k}^B$ and the noise v_k , respectively.

2) *Update using DVL measurements*: The model prediction is updated by the standard Kalman filter equations each time a new DVL measurement arrives:

$$\mathbf{z}_{DVL,k} = [u_b, v_b, w_b, u_w, v_w, w_w, \phi_i, \theta_i, \psi_c, z_{depth}]^T \quad (14)$$

where subindex b stands for bottom tracking velocity, w for through water velocity, i for inclinometers and c represents the compass. The measurement model is:

$$\mathbf{z}_{DVL,k} = \mathbf{H}_{DVL,k} \mathbf{x}_{k|k-1} + w_k \quad (15)$$

$$\mathbf{H}_{DVL} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ 0 & 0 & 1 & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (16)$$

where w_k (measurement noise) is a zero-mean white noise:

$$E[w_k] = 0; \quad E[w_k w_j^T] = \delta_{kj} \mathbf{R}_{DVL,k} \quad (17)$$

Since the DVL sensor provides a status measurement for bottom tracking and water velocity, depending on the quality of the measurements, different versions of the \mathbf{H} matrix are used to fuse one (removing row 2), the other (removing row 1), or both readings (using the full matrix).

3) *Update using MTi measurements*: Whenever a new attitude measurement is available from the MTi sensor, the model prediction is updated using the standard Kalman filter equations:

$$\mathbf{z}_{MTi,k} = [\phi, \theta, \psi]^T, \quad \mathbf{z}_{MTi,k} = \mathbf{H}_{MTi,k} \mathbf{x}_{k|k-1} + w_k \quad (18)$$

$$\mathbf{H}_{MTi} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 6} \end{bmatrix} \quad (19)$$

where w_k (measurement noise) is a zero-mean white noise:

$$E[w_k] = 0; \quad E[w_k w_j^T] = \delta_{kj} \mathbf{R}_{MTi,k} \quad (20)$$

C. Scan forming

The navigation system presented above is able to estimate the robot's pose, but the uncertainty will grow without limit due to its dead-reckoning nature. Moreover, we are only interested in the robot's relative position (and uncertainty) with respect to the beginning of the scan (I). Hence a slight modification to the filter is introduced making a reset in position (setting x, y, z to 0 in the vector state) whenever a new scan is started. Therefore, while the filter is working, the estimated position is always relative to the position where the first beam of the scan was gathered (I). Note that it is important to keep the ψ value (it is not reset) because it represents an absolute angle with respect to the magnetic north and a reset would mean an unreal high rotation during the scan. The same thing happens with ϕ and θ . Since we are only interested in the uncertainty accumulated during the scan, the reset process also affects the x, y , and z terms of the covariance matrix \mathbf{P} . Now, the modified filter provides the robot's relative position where the beams were gathered including its uncertainty accumulated during the scan. Hence, it is possible to reference all the ranges computed from the beams to the initial frame I , removing the distortion induced by the robot's motion by using the following method.

Let

- $\rho \equiv N(\hat{\rho}, \mathbf{P}_\rho)$ be a r.g.v. corresponding to the polar measurement where $\hat{\rho} = (\beta, r)$ is the observed measurement and \mathbf{P}_ρ its corresponding uncertainty.
- $\mathbf{x}_R^B \equiv N(\hat{\mathbf{x}}_R^B, \mathbf{P}_{BR})$ be a r.g.v. corresponding to the robot's uncertain position where the ρ beam was gathered. This value is estimated by the EKF and is represented in the northern referenced frame B .
- $\mathbf{x}_B^I \equiv N(\hat{\mathbf{x}}_B^I, \mathbf{P}_{IB})$ be a r.g.v. corresponding to the transformation needed to map B frame to I frame. In our particular case, this is a null translation followed by a rotation used to align B with I .
- \mathbf{x}_S^R be a deterministic vector that describes the position and attitude of the sensor frame S with respect to the robot's frame R . Note that this is non-random rigid body transformation.

then, it is possible to compute the position (and uncertainty) of any observed point referenced to the initial frame I as follows:

- 1) $\mathbf{p}^S = P2C(\rho) \Rightarrow \mathbf{p}^S = N(\underbrace{P2C(\hat{\rho})}_{\hat{\mathbf{p}}^S}, \underbrace{\mathbf{J}_S \mathbf{P}_\rho \mathbf{J}_S^T}_{\mathbf{P}_S})$
 where $P2C(\rho)$ turn polar into cartesian coordinates and
 $\mathbf{J}_S = \left. \frac{\partial P2C(\rho)}{\partial \rho} \right|_{\hat{\rho}}$
- 2) $\mathbf{p}^R = \mathbf{x}_S^R \oplus \mathbf{p}^S \Rightarrow \mathbf{p}^R = N(\underbrace{\mathbf{x}_S^R \oplus \hat{\mathbf{p}}^S}_{\hat{\mathbf{p}}^R}, \underbrace{\mathbf{J}_{R\oplus} \mathbf{P}_S \mathbf{J}_{R\oplus}^T}_{\mathbf{P}_R})$
 where $\mathbf{J}_{R\oplus} = \left. \frac{\partial \mathbf{x}_S^R \oplus \mathbf{p}^S}{\partial \mathbf{p}^S} \right|_{\hat{\mathbf{p}}^S}$
- 3) $\mathbf{p}^B = \mathbf{x}_R^B \oplus \mathbf{p}^R \Rightarrow \mathbf{p}^B = N(\underbrace{\hat{\mathbf{x}}_R^B \oplus \hat{\mathbf{p}}^R}_{\hat{\mathbf{p}}^B}, \underbrace{\mathbf{J}_{B1\oplus} \mathbf{P}_{BR} \mathbf{J}_{B1\oplus}^T + \mathbf{J}_{B2\oplus} \mathbf{P}_R \mathbf{J}_{B2\oplus}^T}_{\mathbf{P}_B})$
 where $\mathbf{J}_{B1\oplus} = \left. \frac{\partial \mathbf{x}_R^B \oplus \mathbf{p}^R}{\partial \mathbf{x}_R^B} \right|_{\hat{\mathbf{x}}_R^B}$
 and $\mathbf{J}_{B2\oplus} = \left. \frac{\partial \mathbf{x}_R^B \oplus \mathbf{p}^R}{\partial \mathbf{p}^R} \right|_{\hat{\mathbf{p}}^R}$
- 4) $\mathbf{p}^I = \mathbf{x}_B^I \oplus \mathbf{p}^B \Rightarrow \mathbf{p}^I = N(\underbrace{\hat{\mathbf{x}}_B^I \oplus \hat{\mathbf{p}}^B}_{\hat{\mathbf{p}}^I}, \underbrace{\mathbf{J}_{I1\oplus} \mathbf{P}_{IB} \mathbf{J}_{I1\oplus}^T + \mathbf{J}_{I2\oplus} \mathbf{P}_B \mathbf{J}_{I2\oplus}^T}_{\mathbf{P}_I})$
 where $\mathbf{J}_{I1\oplus} = \left. \frac{\partial \mathbf{x}_B^I \oplus \mathbf{p}^B}{\partial \mathbf{x}_B^I} \right|_{\hat{\mathbf{x}}_B^I}$
 and $\mathbf{J}_{I2\oplus} = \left. \frac{\partial \mathbf{x}_B^I \oplus \mathbf{p}^B}{\partial \mathbf{p}^B} \right|_{\hat{\mathbf{p}}^B}$

First, the function $P2C$ transforms the range and bearing data $\rho = (\beta, r)^T$ from Polar space to Cartesian space. The result is the observed point \mathbf{p}^S referenced to the S frame. As stated, \mathbf{p}^S is a r.g.v. which mean ($\hat{\mathbf{p}}^S$) and covariance (\mathbf{P}_S) can be easily computed. Then, by means of a rigid body transformation, the point is referenced to the robot's frame R . Again, the new representation \mathbf{p}^R is a r.g.v. with mean $\hat{\mathbf{p}}^R$ and covariance \mathbf{P}_R . Now, the robot's relative position \mathbf{x}_R^B computed with the EKF is compounded with the robot's referenced point \mathbf{p}^R to get the r.g.v. \mathbf{p}^B with mean $\hat{\mathbf{p}}^B$ and covariance \mathbf{P}_B . Finally, the last compounding operation rotates the point to reference it to the initial frame I . As in

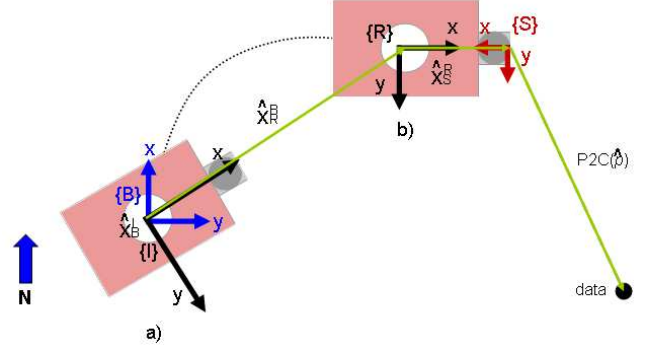


Fig. 8. Scan forming process. Point a represents the position of the robot at the first beam of the scan, point b) Represents it at the position of the beam k.

the previous cases, \mathbf{p}^I is a r.g.v. with a known mean ($\hat{\mathbf{p}}^I$) and covariance \mathbf{P}_I . Fig. 8 illustrates this process while the scan grabbing process in algorithmic notation is described in Algorithm 3.

Algorithm 3 Scan grabbing

```

[ $S_{new}, \hat{\mathbf{q}}_{new}, \mathbf{P}_{q_{new}}$ ] = ScanGrabbing() {
  ResetDeadReckoningXYZ()
  [ $\hat{\mathbf{x}}_B^I, \mathbf{P}_{IB}$ ] = GetDeadReckoning()
  for all beams {
    beam = GetBeam()
    beam = Segment(beam)
    [ $\hat{\rho}, \mathbf{P}_\rho$ ] = LocalMaximaFinder(beam)
    [ $\hat{\mathbf{x}}_R^B, \mathbf{P}_{BR}$ ] = GetDeadReckoning()
    //  $\hat{\rho}$  and  $\mathbf{P}_\rho$  from the local frame  $I$ 
     $\hat{\mathbf{n}} = \hat{\mathbf{x}}_B^I \oplus \hat{\mathbf{x}}_R^B \oplus \mathbf{x}_S^R \oplus P2C(\hat{\rho})$ 
     $\mathbf{P}_n = \mathbf{J}_{I1\oplus} \mathbf{P}_{IB} \mathbf{J}_{I1\oplus}^T + \mathbf{J}_{I2\oplus} [\mathbf{J}_{B1\oplus} \mathbf{P}_{BR} \mathbf{J}_{B1\oplus}^T + \mathbf{J}_{B2\oplus} \mathbf{J}_{R\oplus} \mathbf{J}_S \mathbf{P}_\rho \mathbf{J}_S^T \mathbf{J}_{R\oplus}^T \mathbf{J}_{B2\oplus}^T] \mathbf{J}_{I2\oplus}^T +$ 
     $\mathbf{J}_{B2\oplus} \mathbf{J}_{R\oplus} \mathbf{J}_S \mathbf{P}_\rho \mathbf{J}_S^T \mathbf{J}_{R\oplus}^T \mathbf{J}_{B2\oplus}^T$ 
     $S_{new} = S_{new} \cup \{[\hat{\mathbf{n}}, \mathbf{P}_n]\}$ 
  }
   $\hat{\mathbf{q}}_{new} = \hat{\mathbf{x}}_R^B$ 
   $\mathbf{P}_{q_{new}} = \mathbf{P}_{BR}$ 
}
```

VI. THE MSISPIC ALGORITHM

Once the pIC and the ScanGrabbing algorithms have been setup, it is very simple to localize the robot. This is the purpose of the MSISpIC algorithm (see Algorithm 4), which iteratively grabs two scans and register them using the pIC algorithm. It is worth noting that the pIC takes as input two consecutive scans (S_{new} and S_{ref}) and its relative displacement which coincides with the position occupied by the robot at the end of the first scan ($\hat{\mathbf{q}}_{ref}$). The output is an improved estimation of the robot displacement ($\hat{\mathbf{q}}_{new}$). The iterative compounding of the relative displacement allows to track the global robot position.

VII. EXPERIMENTAL RESULTS

The MSISpIC algorithm has been used with a dataset obtained in an abandoned marina located in Sant Pere Pescador,

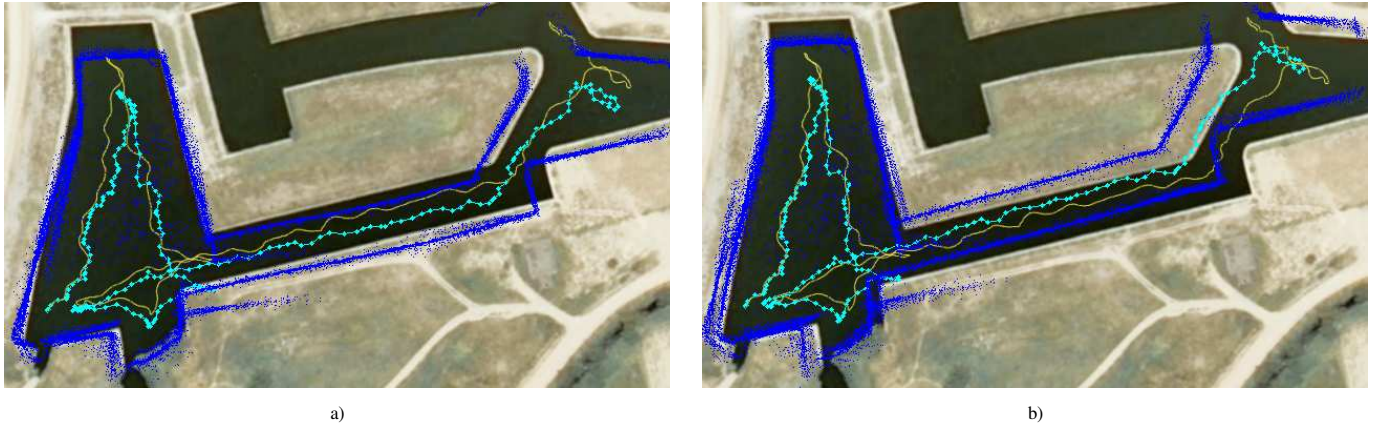


Fig. 10. Results: a) Map and trajectory generated with the MSISpIC output b) Map and trajectory corrected using the translation from MSISpIC output and rotation from the EKF updated with the MTi heading sensor. Comparison with GPS trajectory (yellow).

Algorithm 4 MSISpIC

```

MSISpIC() {
   $[S_{ref}, \hat{\mathbf{q}}_{ref}, \mathbf{P}_{q_{ref}}] = ScanGrabbing()$ 
   $\hat{\mathbf{q}}_{global} = \mathbf{0}$ 
  while(true) {
     $[S_{new}, \hat{\mathbf{q}}_{new}, \mathbf{P}_{q_{new}}] = ScanGrabbing()$ 
     $\hat{\mathbf{q}}_{pIC} = pIC(S_{ref}, S_{new}, \hat{\mathbf{q}}_{ref}, \mathbf{P}_{q_{ref}})$ 
     $\hat{\mathbf{q}}_{global} = \hat{\mathbf{q}}_{global} \oplus \hat{\mathbf{q}}_{pIC}$ 
     $S_{ref} = S_{new}$ 
     $\hat{\mathbf{q}}_{ref} = \hat{\mathbf{q}}_{new}$ 
  }
}

```



Fig. 9. Trajectory and map generated with odometry (cyan). GPS trajectory (yellow) used as a ground truth.

on the Catalan coast [18] [19]. This dataset is useful to test if the algorithm is capable to register the limited information provided by each scan in a large structure. The survey mission was carried out using ICTINEU^{AUV} [8] traveling along a 600 meters path. The MSIS was configured to scan the whole 360° sector and it was set to fire up to a 50m range with a 0.1m resolution and a 1.8° angular step. Dead-reckoning was computed using the velocity reading coming from the

DVL and the heading data obtained from the MTi sensor, both merged using the described EKF. Standard deviation for the MSIS sensor was set as it is specified by the manufacturer, 0.1m in range and 1.8° in angular measurements. Fig. 9 shows the trajectory and the map estimated using the dead-reckoning method. Fig. 10 shows the trajectory and the map estimated with the MSISpIC algorithm. In these figures, the estimated trajectory is plotted on an ortophotomap together with the GPS ground truth for comparison. It can be appreciated that the dead-reckoning estimated trajectory suffers from an important drift which is considerably reduced when the MSISpIC algorithm is used. While the dead-reckoning map (Fig. 9) has worse resolution showing thicker walls, the one build with MSISpIC presents narrower and better defined walls. Both of them show duplicated walls since this is a consequence of the drift which is improved but not avoided with the MSISpIC. In Fig. 10.a it can be appreciated that the mapped size of the polygonal channel is smaller than the actual size. The same happens with the long, almost horizontal, water channel. This problem arises because during part of the trajectory, the robot traverses an area where the scan only observes one or two walls parallel to the robot path, being able to correct the lateral displacement but still drifting in the forward direction. It is worth noting that, even in the presence of structures in all the directions, scan matching algorithms are expected to drift due to its iterative formulation. Although not appreciated in Fig. 10.a, the drift in orientation is particularly dangerous since it quickly propagates to the position. To overcome this problem a variation of the algorithm has been tested. In this case, the MSISpIC estimates translation only while the EKF filter estimates the heading acquired from the heading sensor. The resulting map of such strategy is shown in Fig. 10.b. In this case, the size of the polygonal water channel is more close to the real one and the trajectory seems to be more close to the ground truth estimated with the GPS. Nevertheless, the experiment is not long enough to show a clear advantage.

VIII. CONCLUSIONS

This paper analyzes and compares two well known scan matching algorithms: the MbICP and the pIC. The two algorithms are tested with synthetic noisy data simulating a

post-processed output of a MSIS sensor in a static scenario. The study concludes that using unprecise sonar sensor data the statistical framework of the pIC algorithm provides much better results than the MbICP, which works in a metric framework. Next, a variation of the pIC algorithm called MSISpIC is presented, which is able to perform underwater scan matching using a MSIS. To deal with the motion induced distortion of the acoustic image, an EKF is used to estimate the robot motion during the scan. The filter uses a constant velocity model with acceleration noise for motion prediction and velocity (DVL) and attitude measurements (MTi) for updating the state. Through the compounding of the relative robot position within the scan, with the range and bearing measurements of the beams gathered with the sonar, the acoustic image gets undistorted. Assuming Gaussian noise, the algorithm is able to predict the uncertainty of the sonar measurements with respect to a frame located at the position occupied by the robot at the beginning of the scan, before applying the standard pIC algorithm. The proposed method has been tested with a dataset acquired during a survey mission in an abandoned marina located in the Girona coast. The results show substantial improvements in trajectory correction and map reconstruction.

ACKNOWLEDGMENTS

This research was sponsored by the Spanish government (DPI2005-09001-C03-01) and FREEsubNET (MRTN-CT-2006-036186). The authors want to thank Javier Mínguez and Luís Montesano from University of Zaragoza for their helpful advices to fulfill this paper.

REFERENCES

- [1] C. WangThorpe and A. Suppe, "Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds," *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pp. 416–421, 9–11 June 2003.
- [2] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, pp. 206–211 vol.1, 27–31 Oct. 2003.
- [3] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila, "Autonomous rover navigation on unknown terrains functions and integration," in *ISER '00: Experimental Robotics VII*. London, UK: Springer-Verlag, 2001, pp. 501–510.
- [4] J. Minguez, "Integration of planning and reactive obstacle avoidance in autonomous sensor-based navigation," *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2486–2492, 2–6 Aug. 2005.
- [5] A. Grossmann and M. Poli, "Robust mobile robot localisation from sparse and noisy proximity readings," 1999. [Online]. Available: citeseer.ist.psu.edu/article/gro99robust.html
- [6] I. Cox, "Blanche-an experiment in guidance and navigation of an autonomous robot vehicle," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 2, pp. 193–204, Apr 1991.
- [7] J. A. Castellanos, J. D. Tardos, and J. Neira, "Constraint-based mobile robotlocalization," *Advanced Robotics and Intelligent Systems, IEE*, 1996.
- [8] D. Ribas, N. Palomer, P. Ridao, M. Carreras, and E. Hernandez, "Ictineu AUV wins the first SAUC-E competition," in *IEEE International Conference on Robotics and Automation*, 2007.
- [9] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*
- [10] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152, 2001.
- [11] F. Lu and E. Milios, "Robot pose estimation in unknown environments by matching 2d range scans," in *CVPR94*, 1994, pp. 935–938.
- [12] J. Minguez, F. Lamiriaux, and L. Montesano, "Metric-based scan matching algorithms for mobile robot displacement estimation," *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 3557–3563, 18–22 April 2005.
- [13] L. Montesano, J. Minguez, and L. Montano, "Probabilistic scan matching for motion estimation in unstructured environments," *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 2–6 Aug. 2005.
- [14] A. Burguera, Y. Gonzalez, and G. Oliver, "Probabilistic sonar scan matching for robust localization," *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3154–3160, 10–14 April 2007.
- [15] Y. Bar-Shalom and T. Fortman, *Tracking and Data Association*. Academic Press, 1998.
- [16] D. Ribas, J. Neira, P. Ridao, and J. D. Tardos, "Auv localization in structured underwater environments using an a priori map," in *7th IFAC Conference on Manoeuvring and Control of Marine Crafts*, Lisbon, Portugal, september 2006.
- [17] T. Fossen, *Guidance and Control of Ocean Vehicles*, J. Wiley and S. Ltd., Eds., 1994.
- [18] D. Ribas, "David ribas homepage," <http://eia.udg.edu/~dribas>, Online August 2008.
- [19] D. Ribas, P. Ridao, J. Tardós, and J. Neira, "Underwater SLAM in man made structured environments," *Journal of Field Robotics*, vol. Accepted for publication, 2008.